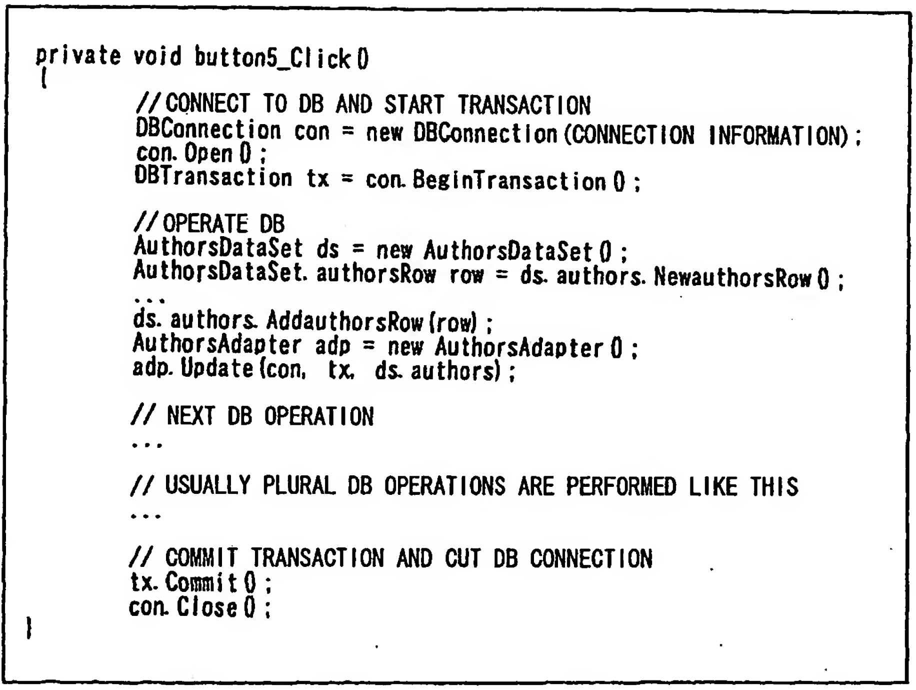
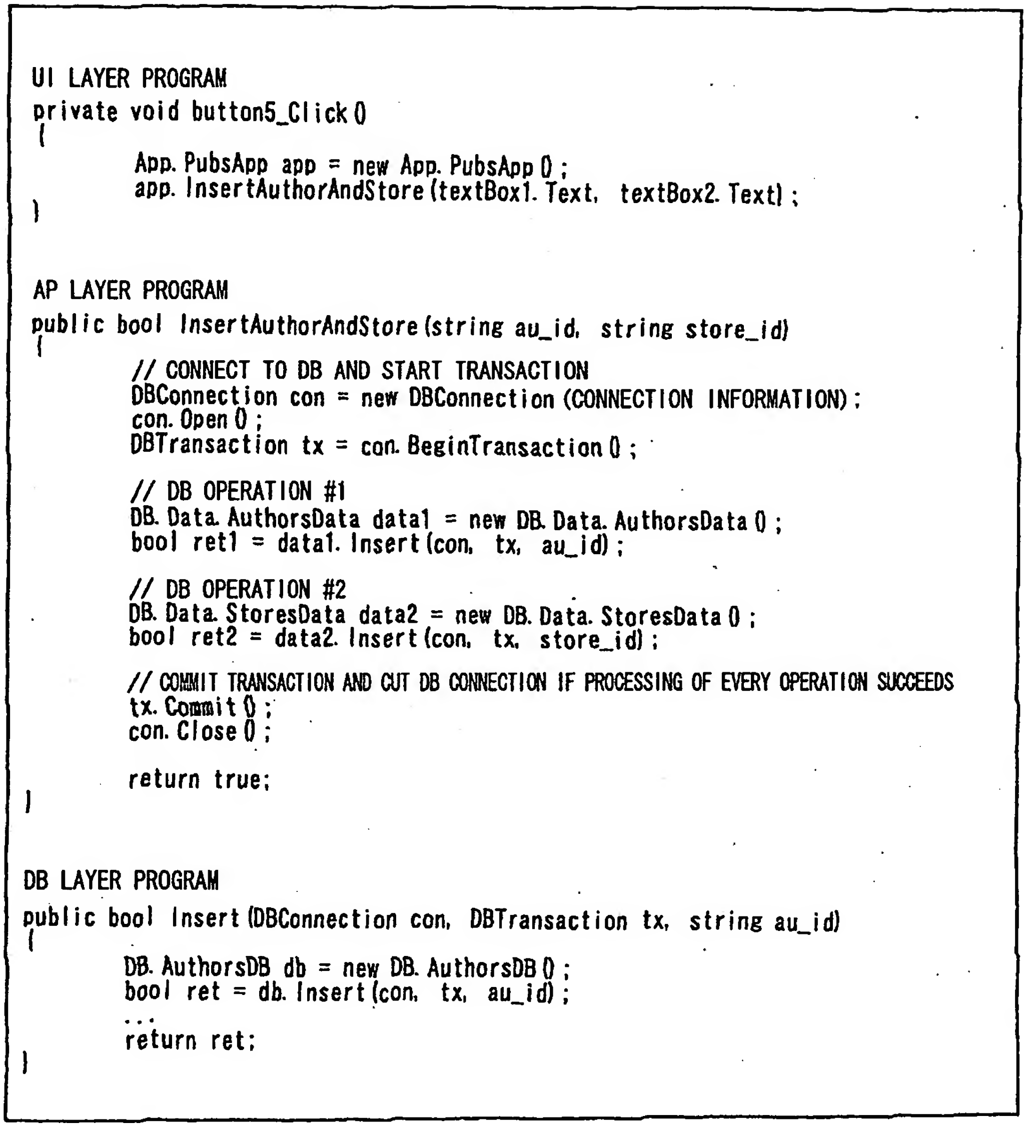
FIG. 1



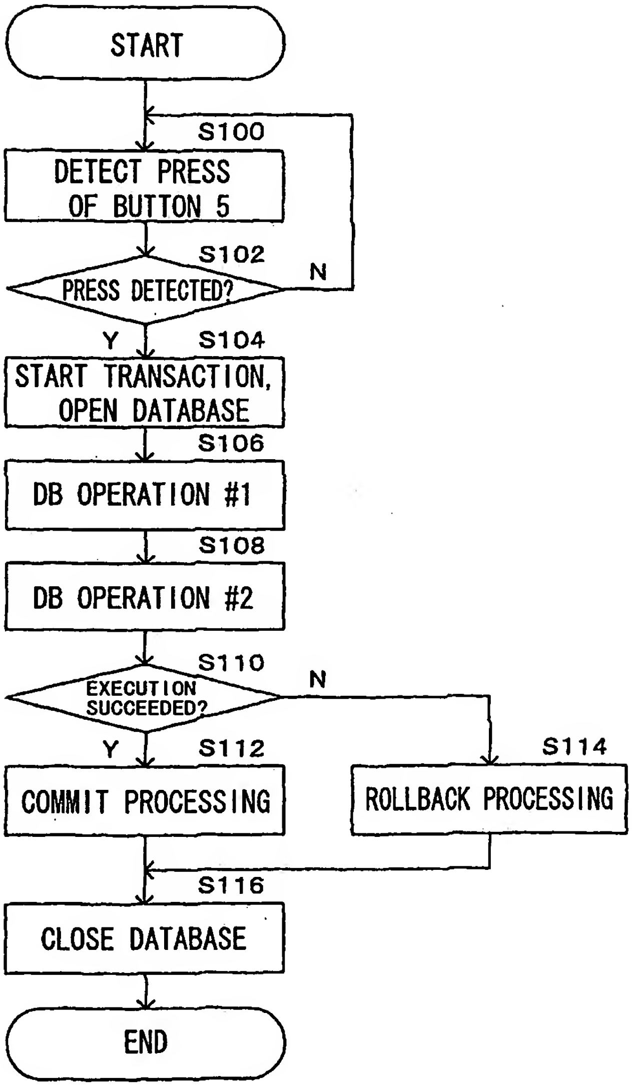
```
private void button5_Click()
{
        //CONNECT TO DB AND START TRANSACTION
        DBConnection con = new DBConnection(CONNECTION INFORMATION);
        con.Open();
        DBTransaction tx = con.BeginTransaction();

        //OPERATE DB
        AuthorsDataSet ds = new AuthorsDataSet();
        AuthorsDataSet.authorsRow row = ds.authors.NewauthorsRow();
        ...
        ds.authors.AddauthorsRow(row);
        AuthorsAdapter adp = new AuthorsAdapter();
        adp.Update(con, tx, ds.authors);

        // NEXT DB OPERATION
        ...

        // USUALLY PLURAL DB OPERATIONS ARE PERFORMED LIKE THIS
        ...

        // COMMIT TRANSACTION AND CUT DB CONNECTION
        tx.Commit();
        con.Close();
}
```

FIG. 2



```
UI LAYER PROGRAM
private void button5_Click()
{
        App.PubsApp app = new App.PubsApp();
        app.InsertAuthorAndStore(textBox1.Text, textBox2.Text);
}

AP LAYER PROGRAM
public bool InsertAuthorAndStore(string au_id, string store_id)
{
        // CONNECT TO DB AND START TRANSACTION
        DBConnection con = new DBConnection(CONNECTION INFORMATION);
        con.Open();
        DBTransaction tx = con.BeginTransaction();

        // DB OPERATION #1
        DB.Data.AuthorsData data1 = new DB.Data.AuthorsData();
        bool ret1 = data1.Insert(con, tx, au_id);

        // DB OPERATION #2
        DB.Data.StoresData data2 = new DB.Data.StoresData();
        bool ret2 = data2.Insert(con, tx, store_id);

        // COMMIT TRANSACTION AND CUT DB CONNECTION IF PROCESSING OF EVERY OPERATION SUCCEEDS
        tx.Commit();
        con.Close();

        return true;
}

DB LAYER PROGRAM
public bool Insert(DBConnection con, DBTransaction tx, string au_id)
{
        DB.AuthorsDB db = new DB.AuthorsDB();
        bool ret = db.Insert(con, tx, au_id);
        ...
        return ret;
}
```

FIG. 3


START
S100
DETECT PRESS OF BUTTON 5
S102
PRESS DETECTED?
N
Y
S104
START TRANSACTION, OPEN DATABASE
S106
DB OPERATION #1
S108
DB OPERATION #2
S110
EXECUTION SUCCEEDED?
N
Y
S112
COMMIT PROCESSING
S114
ROLLBACK PROCESSING
S116
CLOSE DATABASE
END

S10

FIG. 4

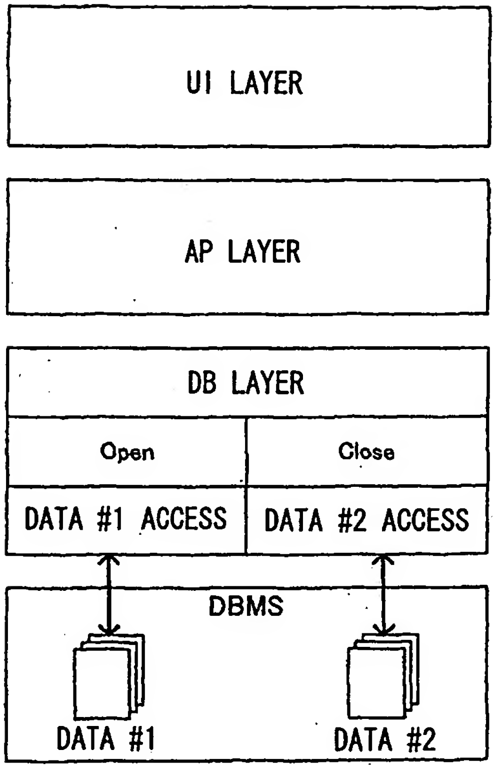
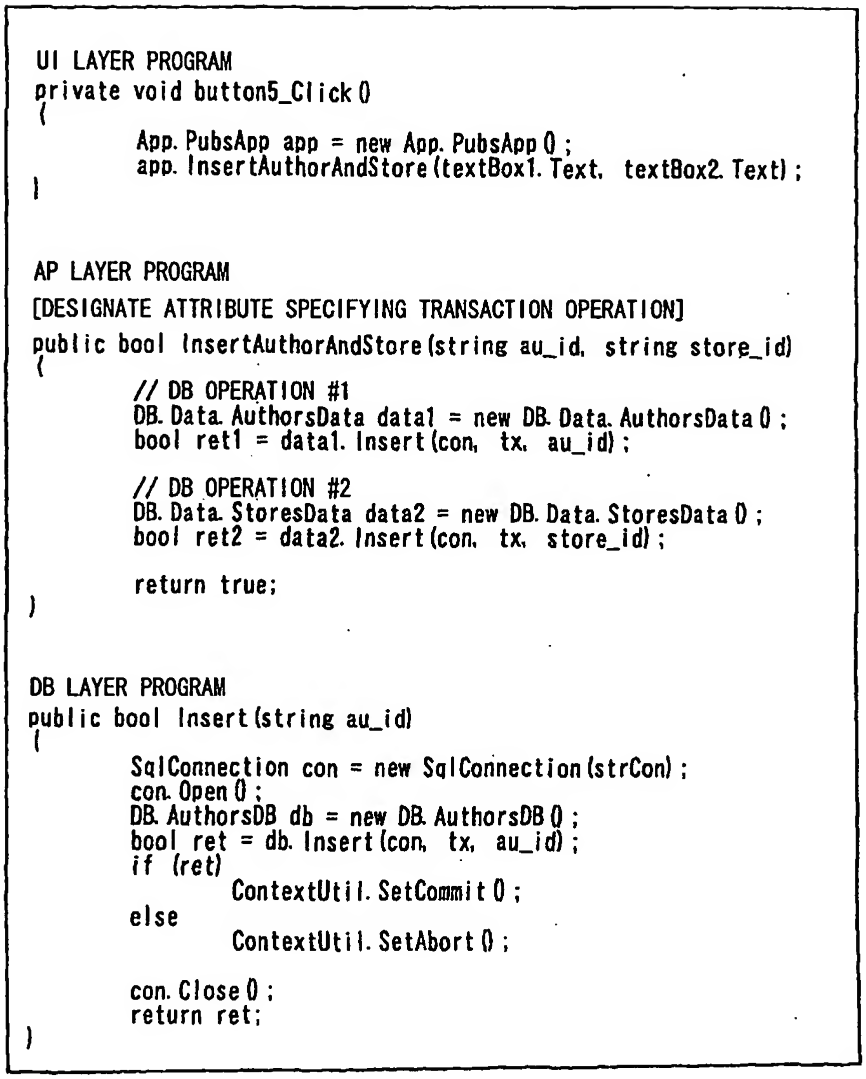

UI LAYER
AP LAYER
DB LAYER
Open
Close
DATA #1 ACCESS
DATA #2 ACCESS
DBMS
DATA #1
DATA #2

FIG. 5



```
UI LAYER PROGRAM
private void button5_Click()
{
        App.PubsApp app = new App.PubsApp();
        app.InsertAuthorAndStore(textBox1.Text, textBox2.Text);
}


AP LAYER PROGRAM
[DESIGNATE ATTRIBUTE SPECIFYING TRANSACTION OPERATION]
public bool InsertAuthorAndStore(string au_id, string store_id)
{
        // DB OPERATION #1
        DB.Data.AuthorsData data1 = new DB.Data.AuthorsData();
        bool ret1 = data1.Insert(con, tx, au_id);

        // DB OPERATION #2
        DB.Data.StoresData data2 = new DB.Data.StoresData();
        bool ret2 = data2.Insert(con, tx, store_id);

        return true;
}


DB LAYER PROGRAM
public bool Insert(string au_id)
{
        SqlConnection con = new SqlConnection(strCon);
        con.Open();
        DB.AuthorsDB db = new DB.AuthorsDB();
        bool ret = db.Insert(con, tx, au_id);
        if (ret)
                ContextUtil.SetCommit();
        else
                ContextUtil.SetAbort();

        con.Close();
        return ret;
}
```

FIG. 6

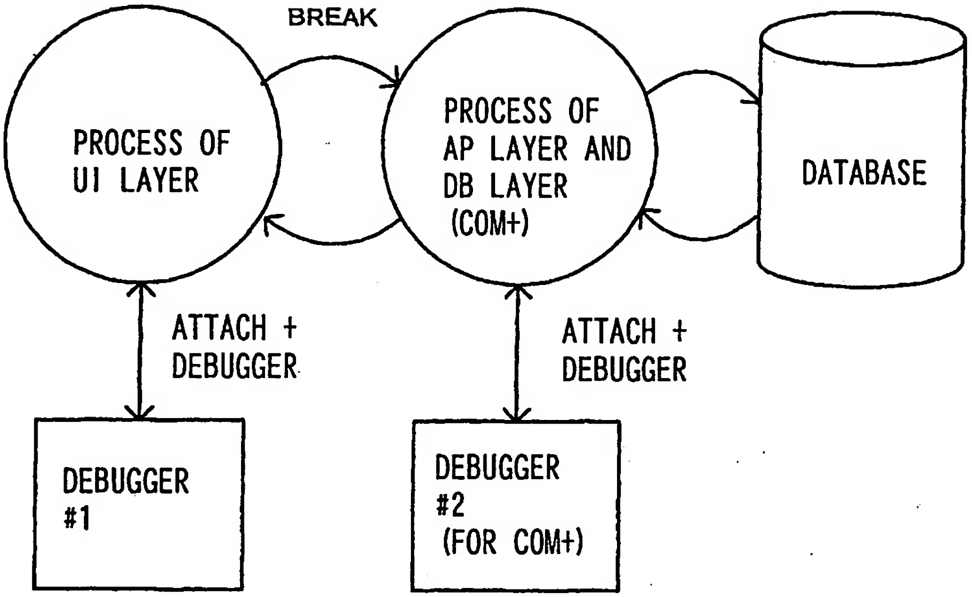

BREAK
PROCESS OF UI LAYER
PROCESS OF AP LAYER AND DB LAYER (COM+)
DATABASE
ATTACH + DEBUGGER
ATTACH + DEBUGGER
DEBUGGER #1
DEBUGGER #2 (FOR COM+)

FIG. 7

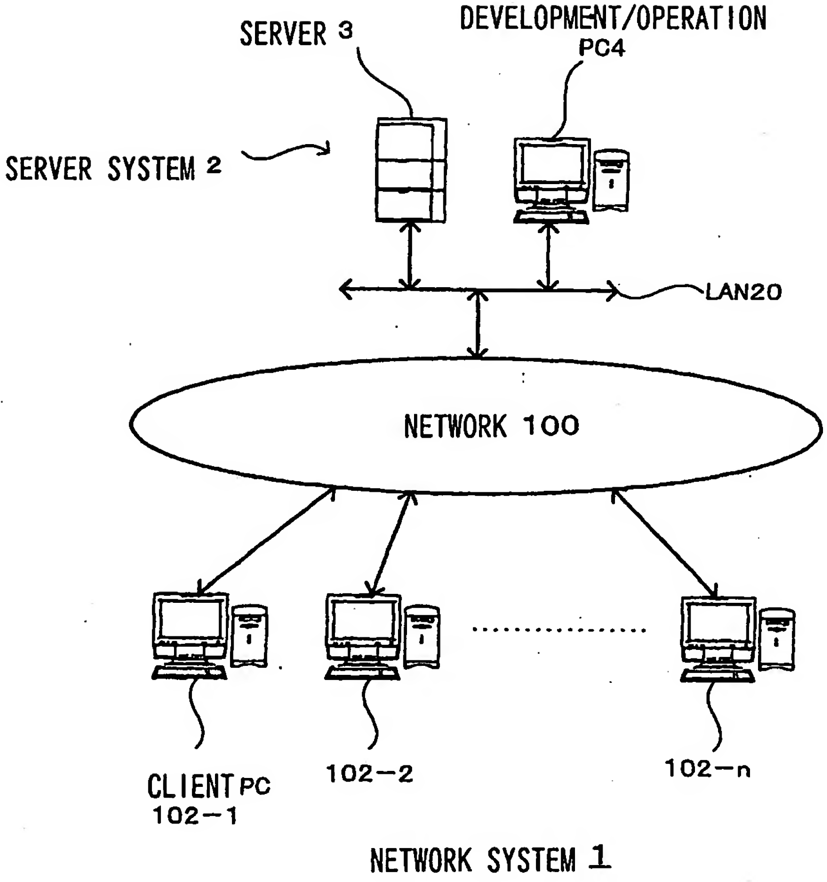
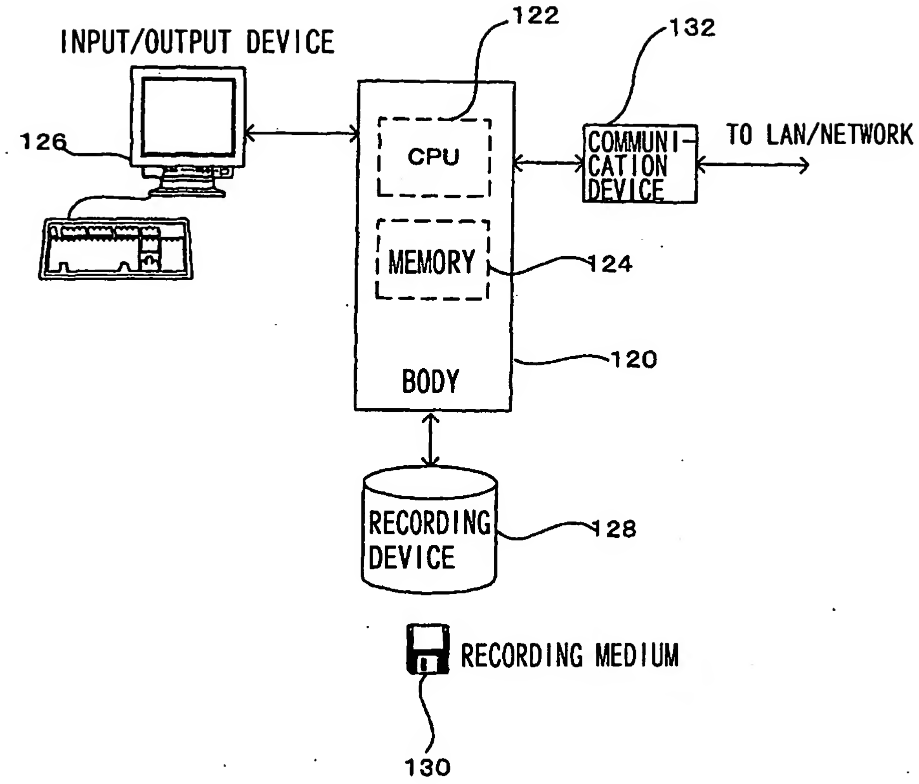
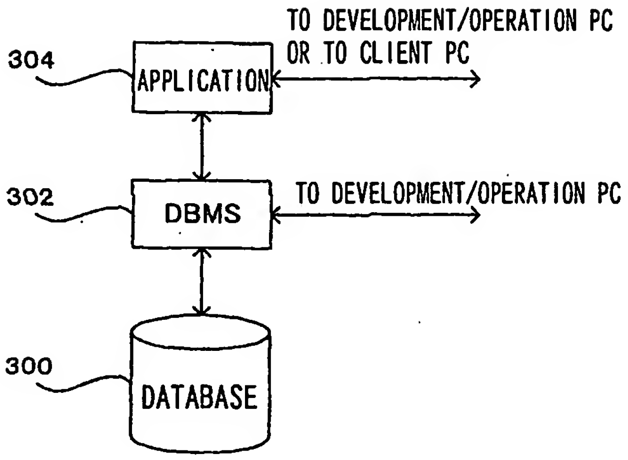

SERVER 3
DEVELOPMENT/OPERATION PC4
SERVER SYSTEM 2
LAN20
NETWORK 100
CLIENT PC 102−1
102−2
102−n
NETWORK SYSTEM 1

FIG. 8



SERVER 3,
PC4, 102

FIG. 9


TO DEVELOPMENT/OPERATION PC OR TO CLIENT PC
304
APPLICATION
302
DBMS
TO DEVELOPMENT/OPERATION PC
300
DATABASE

SERVER PROGRAM 30

FIG. 10

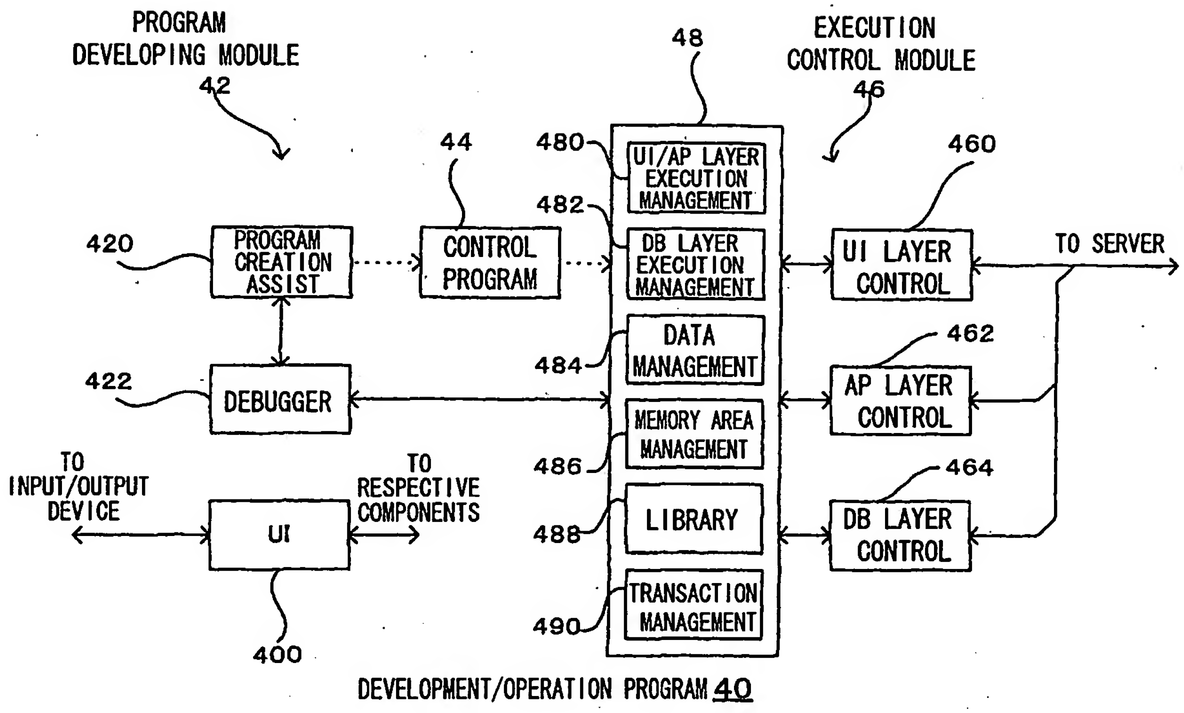
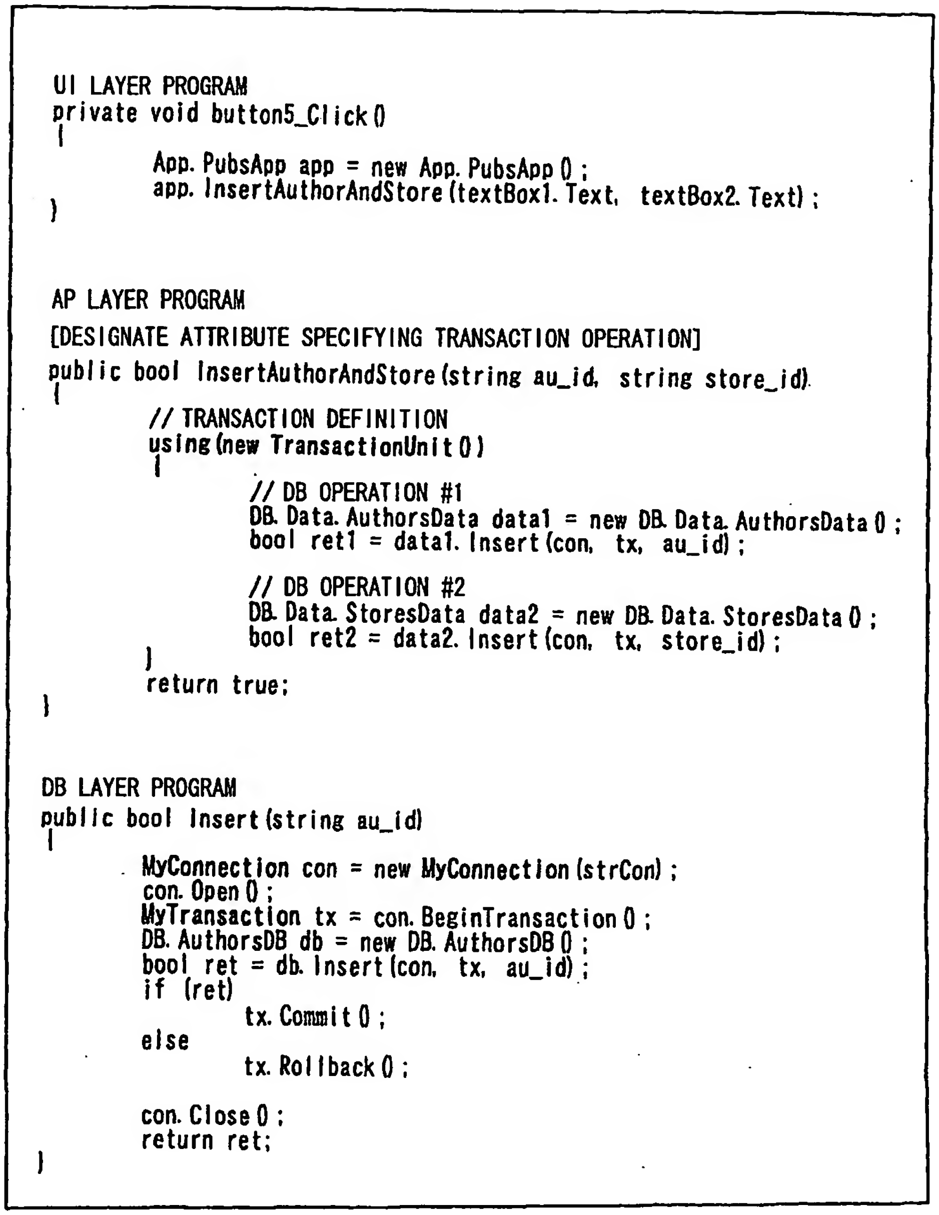

PROGRAM
DEVELOPING MODULE
42
44
48
EXECUTION
CONTROL MODULE
46
480
UI/AP LAYER EXECUTION MANAGEMENT
482
DB LAYER EXECUTION MANAGEMENT
460
420
PROGRAM CREATION ASSIST
CONTROL PROGRAM
UI LAYER CONTROL
TO SERVER
484
DATA MANAGEMENT
462
422
DEBUGGER
AP LAYER CONTROL
MEMORY AREA MANAGEMENT
TO INPUT/OUTPUT DEVICE
TO RESPECTIVE COMPONENTS
486
464
UI
488
LIBRARY
DB LAYER CONTROL
490
TRANSACTION MANAGEMENT
400
DEVELOPMENT/OPERATION PROGRAM 40

FIG. 11



```
UI LAYER PROGRAM
private void button5_Click()
{
        App.PubsApp app = new App.PubsApp();
        app.InsertAuthorAndStore(textBox1.Text, textBox2.Text);
}


AP LAYER PROGRAM
[DESIGNATE ATTRIBUTE SPECIFYING TRANSACTION OPERATION]
public bool InsertAuthorAndStore(string au_id, string store_id)
{
        // TRANSACTION DEFINITION
        using(new TransactionUnit())
        {
                // DB OPERATION #1
                DB.Data.AuthorsData data1 = new DB.Data.AuthorsData();
                bool ret1 = data1.Insert(con, tx, au_id);

                // DB OPERATION #2
                DB.Data.StoresData data2 = new DB.Data.StoresData();
                bool ret2 = data2.Insert(con, tx, store_id);
        }
        return true;
}


DB LAYER PROGRAM
public bool Insert(string au_id)
{
        MyConnection con = new MyConnection(strCon);
        con.Open();
        MyTransaction tx = con.BeginTransaction();
        DB.AuthorsDB db = new DB.AuthorsDB();
        bool ret = db.Insert(con, tx, au_id);
        if (ret)
                tx.Commit();
        else
                tx.Rollback();

        con.Close();
        return ret;
}
```

FIG. 12

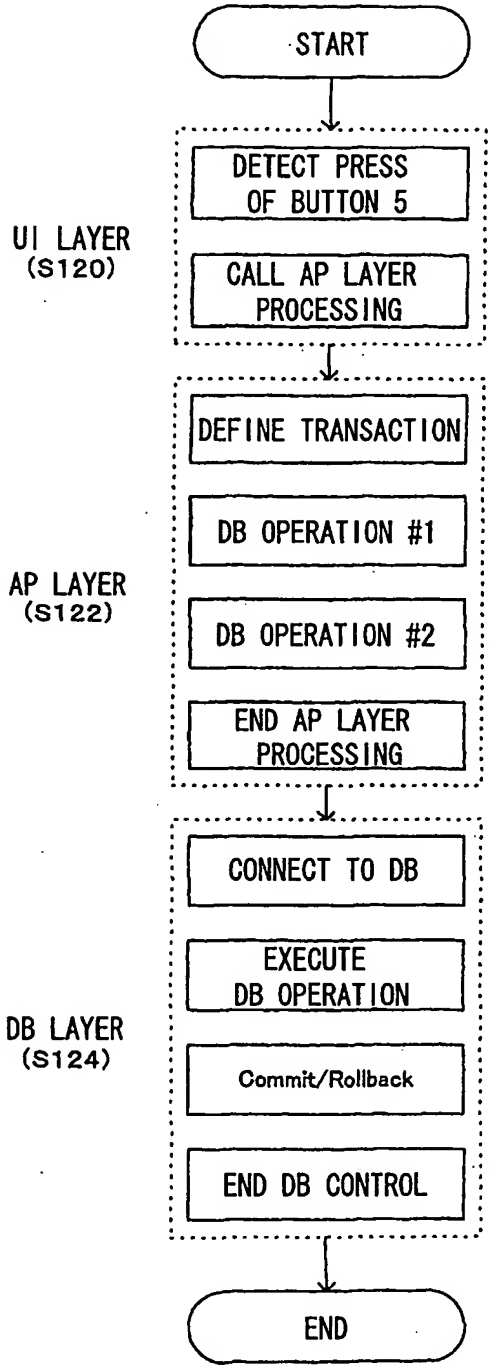

START
DETECT PRESS OF BUTTON 5
CALL AP LAYER PROCESSING
UI LAYER (S120)
DEFINE TRANSACTION
DB OPERATION #1
DB OPERATION #2
END AP LAYER PROCESSING
AP LAYER (S122)
CONNECT TO DB
EXECUTE DB OPERATION
Commit/Rollback
END DB CONTROL
DB LAYER (S124)
END

FIG. 13

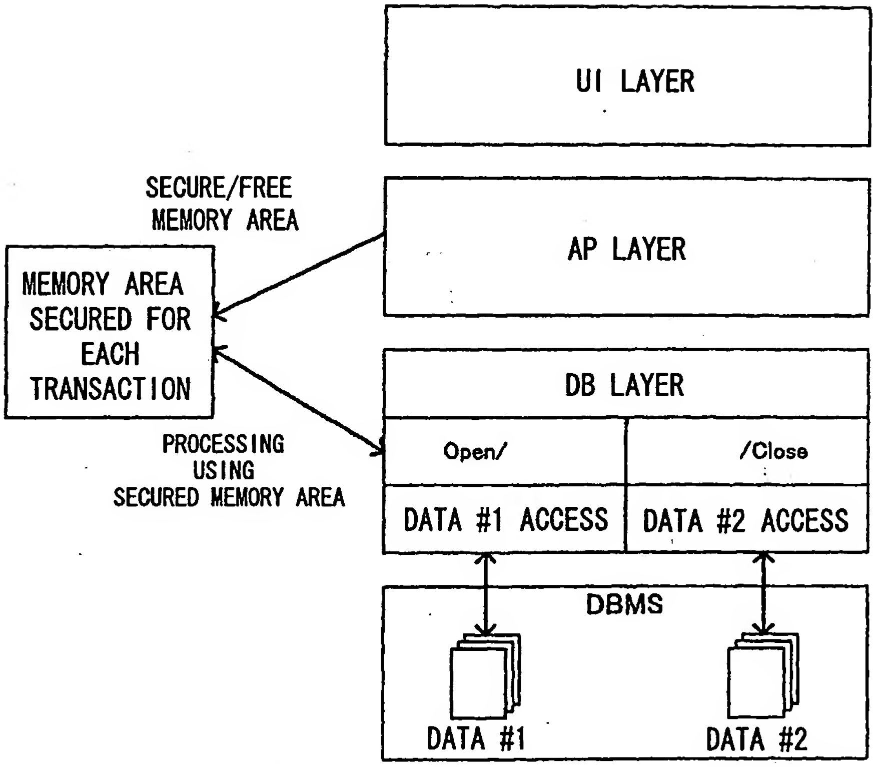

UI LAYER
SECURE/FREE MEMORY AREA
AP LAYER
MEMORY AREA SECURED FOR EACH TRANSACTION
DB LAYER
Open/
/Close
PROCESSING USING SECURED MEMORY AREA
DATA #1 ACCESS
DATA #2 ACCESS
DBMS
DATA #1
DATA #2

FIG. 14

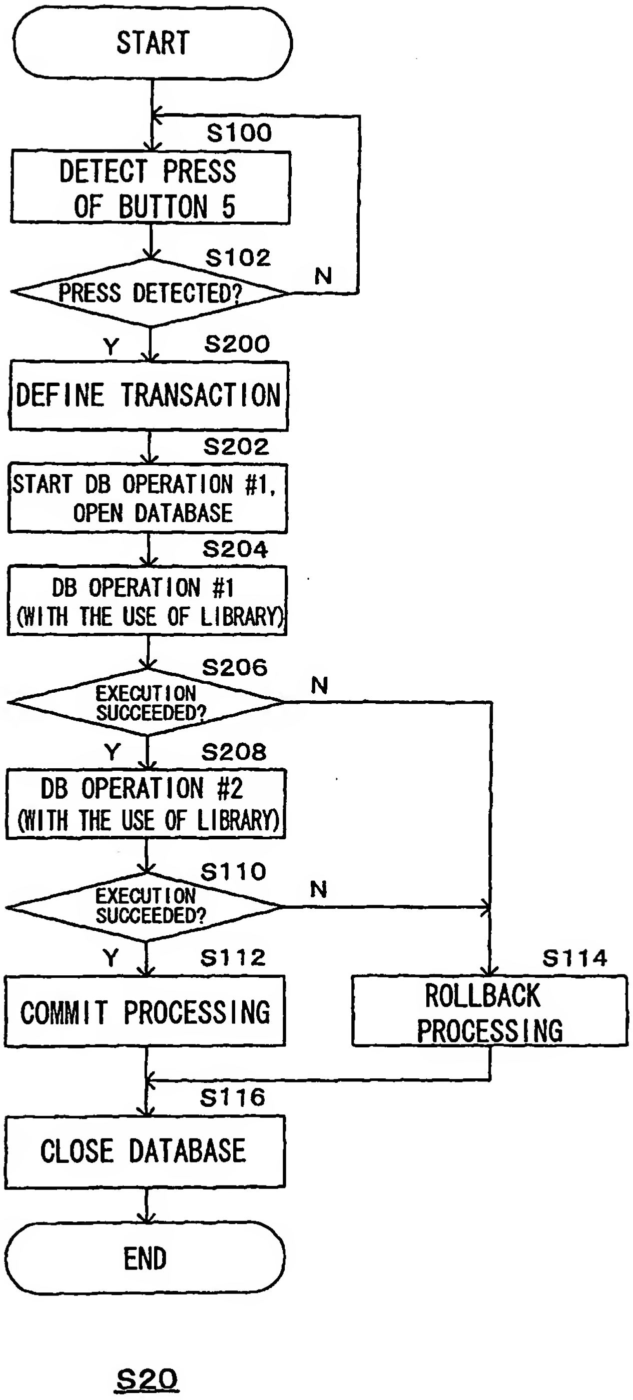

START
S100
DETECT PRESS OF BUTTON 5
S102
PRESS DETECTED?
N
Y
S200
DEFINE TRANSACTION
S202
START DB OPERATION #1, OPEN DATABASE
S204
DB OPERATION #1 (WITH THE USE OF LIBRARY)
S206
EXECUTION SUCCEEDED?
N
Y
S208
DB OPERATION #2 (WITH THE USE OF LIBRARY)
S110
EXECUTION SUCCEEDED?
N
Y
S112
COMMIT PROCESSING
S114
ROLLBACK PROCESSING
S116
CLOSE DATABASE
END
S20

FIG. 15

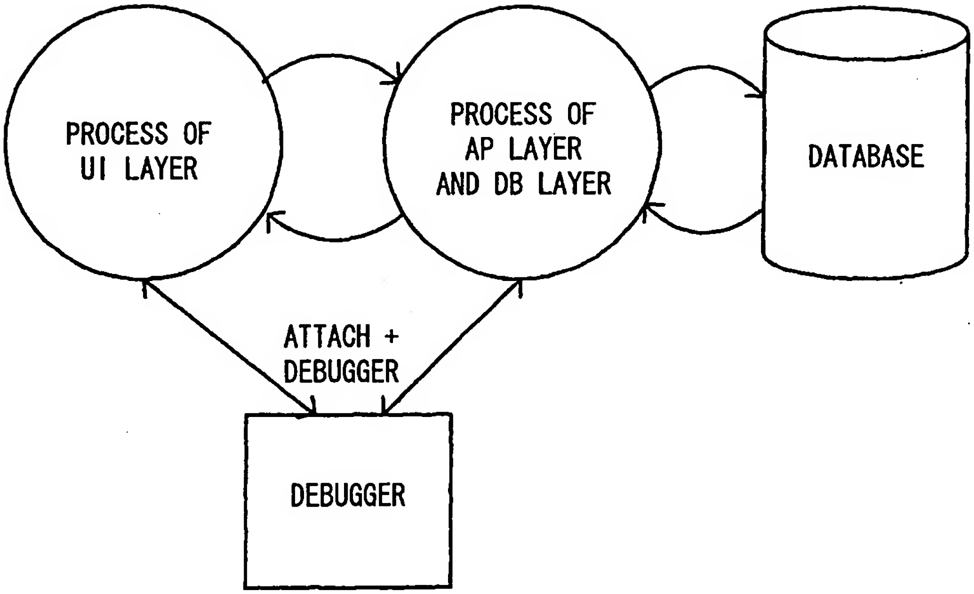
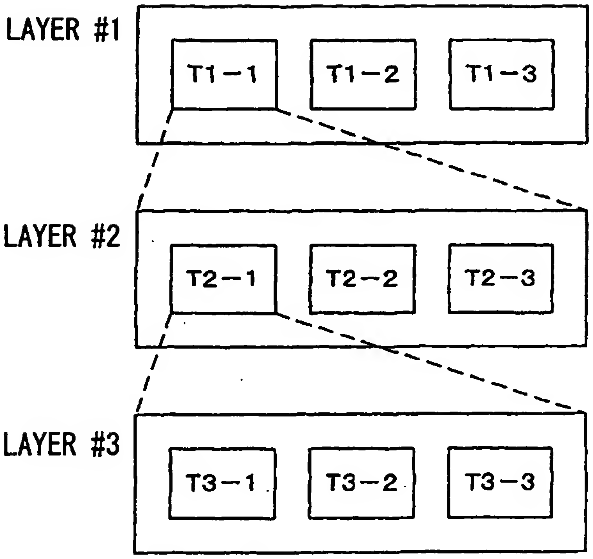

PROCESS OF
UI LAYER
PROCESS OF
AP LAYER
AND DB LAYER
DATABASE
ATTACH +
DEBUGGER
DEBUGGER

FIG. 16



CONTROL PROGRAM 44